

LAB 06

Miss Atika Islam

## 1. Quick Sort

### Overview

Quick Sort is a **divide-and-conquer** algorithm that selects a "pivot" element and partitions the array around the pivot, with elements smaller than the pivot on one side and elements larger on the other.

* **Time Complexity**: Average O(nlogn), Worst )O(n2)
* **Space Complexity**: O(logn) for recursive stack space

### Algorithm

1. Choose a pivot element.
2. Partition the array into two halves around the pivot.
3. Recursively apply the steps to each half.

### Pseudocode

void quickSort(int arr[], int low, int high) {

if (low < high) {

int pivotIndex = partition(arr, low, high);

quickSort(arr, low, pivotIndex - 1);

quickSort(arr, pivotIndex + 1, high);

}

}

int partition(int arr[], int low, int high) {

int pivot = arr[high];

int i = low - 1;

for (int j = low; j < high; j++) {

if (arr[j] < pivot) {

i++;

std::swap(arr[i], arr[j]);

}

}

std::swap(arr[i + 1], arr[high]);

return i + 1;

}

### Practice Problems

1. Implement quick sort for an array of random integers.
2. Modify quick sort to use the first element as the pivot.
3. Sort only the even numbers in ascending order while leaving odd numbers in their original positions.

## Selection Sort

### Overview

Selection Sort is a simple algorithm that repeatedly selects the minimum element from the unsorted portion and moves it to the beginning.

* **Time Complexity**: O(n2)
* **Space Complexity**: O(1)

### Algorithm

1. Find the minimum element in the unsorted portion.
2. Swap it with the first unsorted element.
3. Move the boundary of the sorted portion one element forward.

### Pseudocode

void selectionSort(int arr[], int n) {

for (int i = 0; i < n - 1; i++) {

int min\_index = i;

for (int j = i + 1; j < n; j++) {

if (arr[j] < arr[min\_index]) {

min\_index = j;

}

}

std::swap(arr[i], arr[min\_index]);

}

}

### Practice Problems

1. Implement selection sort for an array of floating-point numbers.
2. Modify selection sort to sort in descending order.
3. Use selection sort to sort an array of strings by length.